Theoretical Computer Science 246 (2000) 107–116

Theoretical Computer Science

www.elsevier.com/locate/tcs

in hypergraphs

Toshihiro Fujito *

Department of Electrical Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima 739-8527, Japan

Received May 1998; revised January 1999
Communicated by C. Ausiello

Abstract

The feedback vertex set problem for hypergraphs is considered and an efficient approximation algorithm is presented. It is shown that an approximation factor of k is guaranteed when the cardinality of every hyperedge is bounded by an integer k, generalizing the existing result for ordinary graphs. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Approximation algorithm; Performance guarantee; Feedback vertex set; Hypergraph

1. Introduction

The *vertex cover* problem is perhaps one of the most well-known basic NP-complete graph problems [12]. For any graph [1] G a set of its vertices is called a *vertex cover* (*VC*) if it contains an end-vertex of every edge of G. In general some weight is associated with each vertex of G, and the weight of a vertex set is defined to be the sum of weights of vertices in the set. The problem is then that of finding a minimum weight VC for G. Given the intractability of the VC problem for exact computation despite its importance in many applications, an efficient and high-quality approximation method for it has been a subject of active research over the years. The guaranteed approximation with a multiplicative factor 2 of the optimum was found early by Gavril [8, p. 134] for the unit cost case, using a simple heuristic based on maximal matching, and many other heuristics with equality good performance guarantee are available today even for the general cost case (see, e.g. [11]). The best performance ratio currently known is $2 - \log\log n/2\log n$ of [3, 17].

* Current Address: Department of Electronics, Nagoya University, Furo Chikusa, Nagoya, 464-8603 Japan. Tel.:/fax: +81-52-789-5337.

E-mail address: fujito@nuee.nagoya-u.ac.jp. (T. Fujito).

[1] This paper deals with only *undirected* (hyper-)graphs.

PII: S0304-3975(99)00043-2

The subject of this paper is a problem closely related to VC, called the *feedback vertex set* problem. A *feedback vertex set* (*FVS*) of a graph G is a set F of vertices s.t. every cycle in G goes through some vertex of F, and the problem is that of finding a minimum weight FVS for a given G. The FVS problem has a distinct history of its own concerning its approximability, initiated by the combinatorial interest in the number of vertex disjoint cycles in a graph [6]. Unlike the case of VC problem, however, a constant factor approximation was found only recently. Bar-Yehuda et al. obtained a factor 4 approximation for the unit cost case, an improvement from the previous best of $\sqrt{\log n}$ [16], but only $4\log n$ factor for general costs [4], which was followed soon afterwards by the improved factor of 2 [2, 5], matching the best constant for the VC problem.

Meanwhile, powerful lower bound techniques have been introduced with the advent of PCP theory in the study of approximation for NP-hard optimization problems. The VC problem was shown MAX SNP-hard in the seminal paper of Papadimitriou and Yannakakis [18] (hence, no polynomial time approximation scheme [1]), and the lower bound on the polynomial time approximation factor for VC has been continuously improved in the last few years; currently it is known to be at least as large as arbitrarily close to 7/6 (modulo $\mathrm{P} \neq \mathrm{NP}$) [9]. As for the FVS problem the VC problem can be reduced to it with no loss of approximation quality [4], and hence, every lower bound on approximability for VC applies to that for FVS.

Allowing not only binary but arbitrary arity relations called hyperedges, a hypergraph provides itself as a more versatile and sometimes only usable model than an ordinary graph, in many applications. It is thus quite natural and important to consider the VC and FVS problems with hypergraphs in their problem domains. In fact, the VC problem for hypergraphs is more commonly termed as the *hitting set* problem, which in turn is known to be equivalent to the *set cover* problem. A k-hypergraph is the one in which the cardinality of every (hyper-)edge is bounded by an integer k. It is then a classic result by now that a factor 2 approximation for VC on ordinary graphs can be generalized to the one with a factor of k on k-hypergraphs [10] (and no better constant factor is known for any k). In a good contrast such a result has not been previously known for the FVS problem.

In this paper we will show that the FVS problem on k-hypergraphs can be approximated with a factor of k. It will be done so by means of the primal–dual algorithm based on an IP formulation of the FVS, using graphic polymatroid functions, which can be thought of as a generalization of the approach for the FVS on ordinary graphs as exhibited in [7]. Notice, as with ordinary graphs, VC on k-hypergraphs can be reduced in an approximation preserving manner to FVS on k-hypergraphs; create a cycle for every existing edge by simply attaching a new edge to it with at least two vertices shared by them. For these reasons any improvement upon the factor obtained in the paper is deemed challenging.

1.1. Notation and definitions

For any hypergraph G let $V(G)$ and $E(G)$ denote the vertex set and the edge set of G, respectively. We treat a (hyper-)edge e of $G=(V,E)$ simply as a set of vertices of G (so $e \subseteq V$). For $X,Y \subseteq V$ let $E[X]$ denote the set of edges induced by X $(=\{e \in E\colon e \subseteq X\})$, and $E[X,Y]$ the set of edges intersected by both X and Y $(=\{e \in E\colon e \cap X \neq \emptyset, e \cap Y \neq \emptyset\})$. A subgraph of G induced by $X \subseteq V$ is denoted by $G[X](=(X,E[X]))$. The set of edges incident to some vertex of X is denoted by $\delta(X)$ and when those edges are restricted to the ones in a subgraph $G[Y]$ we denote it by $\delta_Y(X)(=\delta(X) \cap E[Y])$. Let $\delta(u)$ ($\delta_Y(u)$, resp.) be a shorthand of $\delta(\{u\})$ ($\delta_Y(\{u\})$, resp.).

Any hyperedge e is also represented by the set $\overline{e}$ of ordinary edges, which consists of those connecting any two distinct vertices in e; i.e., $\overline{e}=\{\{u,v\} \subseteq e\colon u \neq v\}$. We assume here that the representations for different hyperedges are disjoint from each other. Thus, whenever two distinct hyperedges $e_1, e_2 \in E$ share at least two vertices, their representations $\overline{e_1}, \overline{e_2}$ introduce multiple (parallel) edges in their union. Extending this notation to a set $F \subseteq E$ of hyperedges we denote by $\overline{F}$ the edge set formed by taking a disjoint union of $\overline{e}$'s for $e \in F$, and $\overline{G}=(V,\overline{E})$ denotes the ordinary graph corresponding to a hypergraph G.

It is customary to measure the quality of an approximation algorithm by its *performance ratio*: the worst-case ratio between the optimal cost and the cost achieved by the algorithm for the same instance.

2. IP Formulation and algorithm

Let $G=(V,E)$ be a k-hypergraph (i.e., $|e| \leqslant k, \forall e \in E$), and $M=(\overline{E},r)$ be the cycle matroid defined on $\overline{G}=(V,\overline{E})$, where r is the rank function of M. We also work on the graphic polymatroid $P=(E,f)$ defined on G; that is, f is the rank function of P defined s.t. $f(F) \stackrel{\text{def}}{=} r(\overline{F})$ for any $F \subseteq E$. The rank of any edge e of G is then given by $f(e)=r(\overline{e})=|e|-1$, one less than its cardinality. Let $\sigma(F)$ denote the sum of ranks of edges in F (i.e., $=\sum_{e \in F} f(e)$). Due to the submodular property of f, we have $f(F) \leqslant \sigma(F)$ for any F, and when the equality holds (i.e., $f(F)=\sigma(F)$), F is called a *matching* of P.

Proposition 1. *Any edge set is acyclic in G iff it is a matching of P.*

Now for any $S \subseteq V$, define another function f_S^{d} on subsets of $E[S]$ s.t.

$$f_S^{\mathrm{d}}(F) \stackrel{\text{def}}{=} \sigma(F) - (f(E[S]) - f(E[S]-F))$$

for $F \subseteq E[S]$, which can be regarded as the dual function of f w.r.t. the subgraph $G[S]$.

Proposition 2. *The set function f_S^{d} is non-decreasing and submodular.*

We claim that the FVS problem on $G=(V,E)$ with weight w_u on $u\in V$ can be formulated by the following integer program:

$$
\text{(IP)}\qquad
\begin{aligned}
\text{Min}\quad & \sum_{u\in V} w_u x_u\\
\text{s.t.}\quad & \\
& \sum_{u\in S} f_S^{\mathrm{d}}(\delta_S(u))x_u \geqslant f_S^{\mathrm{d}}(E[S]), \quad S\subseteq V,\\
& x_u\in\{0,1\}, \quad u\in V.
\end{aligned}
$$

Theorem 3. *A set $F\subseteq V$ is an FVS in G iff $x^F\in\{0,1\}^V$ (incidence vector of F) is a feasible solution to* (IP).

Proof. Notice first that any $X\subseteq V$ is an FVS in G iff $X\cap S$ is an FVS in $G[S]$ for all $S\subseteq V$. Take any S and consider the induced subgraph $G[S]$. Then, since an edge subset of $G[S]$ is acyclic iff it is a matching of P,

$$
\begin{aligned}
Y\subseteq S \text{ is an FVS in } G[S] &\Leftrightarrow E[S]-\delta_S(Y) \text{ is a matching of } P\\
&\Leftrightarrow f(E[S]-\delta_S(Y))=\sigma(E[S]-\delta_S(Y))\\
&\qquad =\sigma(E[S])-\sigma(\delta_S(Y))\\
&\Leftrightarrow \sigma(\delta_S(Y))-(f(E[S])-f(E[S]-\delta_S(Y)))\\
&\qquad =\sigma(E[S])-f(E[S])\\
&\Leftrightarrow f_S^{\mathrm{d}}(\delta_S(Y))=f_S^{\mathrm{d}}(E[S]).
\end{aligned}
$$

Thus, X is an FVS in G iff $f_S^{\mathrm{d}}(\delta_S(X\cap S))=f_S^{\mathrm{d}}(E[S])$ for all $S\subseteq V$.

Suppose first that F is an FVS in G. Because f_S^{d} is a non-negative submodular function $\sum_i f_S^{\mathrm{d}}(E_i)\geqslant f_S^{\mathrm{d}}(\bigcup_i E_i)$ for any family of E_i's, $E_i\subseteq E[S]$, and hence,

$$
\sum_{u\in S} f_S^{\mathrm{d}}(\delta_S(u))x_u^F=\sum_{u\in F\cap S} f_S^{\mathrm{d}}(\delta_S(u))\geqslant f_S^{\mathrm{d}}(\delta_S(F\cap S))=f_S^{\mathrm{d}}(E[S])
$$

for all $S\subseteq V$. Suppose next that F is not an FVS in G, which implies that there exists $S\subseteq V$ s.t. $f_S^{\mathrm{d}}(\delta_S(F\cap S)) < f_S^{\mathrm{d}}(E[S])$. But then,

$$
\begin{aligned}
0 &< f_S^{\mathrm{d}}(E[S])-f_S^{\mathrm{d}}(\delta_S(F\cap S))\\
&=(\sigma(E[S])-f(E[S]))-(\sigma(\delta_S(F\cap S))-(f(E[S])-f(E[S]-\delta_S(F\cap S))))\\
&=\sigma(E[S]-\delta_S(F\cap S))-f(E[S]-\delta_S(F\cap S))\\
&=f_{S-F}^{\mathrm{d}}(E[S-F])
\end{aligned}
$$

Input: a hypergraph $G=(V,E)$ with vertex weights $w_u \geqslant 0$
Output: a feedback vertex set F

```
Initialize F = ∅, S' = V, y = 0, l = 0.
While F is not an FVS in G do
/* invariant: S' = V − F */
    l ← l + 1.
    Increase y_{S'} until for some u ∈ S' the dual constraint (of (D))
        corresponding to u becomes tight.
    Let u_l ← u.
    Add u_l into F and remove u from S'.
For j = l downto 1 do
    if F − {u_j} is an FVS in G then remove u_j from F.
Output F.
```

Fig. 1. Primal–dual algorithm PD.

since $E[S]-\delta_S(F\cap S)=E[S-F]$. Clearly $\sum_{u\in S-F} f^{\mathrm{d}}_{S-F}(\delta_{S-F}(u))x^F_u=0$, and thus x^F does not satisfy the constraint corresponding to $S-F$. $\square$

Take the LP relaxation of (IP) by replacing every integrality constraint on x_u by $x_u \geqslant 0$. Then its dual is

$$\text{(D)}\qquad \begin{aligned} &\text{Max} \quad \sum_{S\subseteq V} f^{\mathrm{d}}_S(E[S])y_S \\ &\text{s.t.} \\ &\qquad \sum_{S:u\in S} f^{\mathrm{d}}_S(\delta_S(u))y_S \leqslant w_u, \quad u\in V, \\ &\qquad y_S \geqslant 0, \quad S\subseteq V. \end{aligned}$$

The primal–dual algorithm PD, presented in Fig. 1, is designed based on (IP) and (D). The algorithm PD starts with $F=\emptyset$, the original graph $G[S']=(V,E)$ and the dual feasible solution $y=0$. Given F, if it is not yet an FVS in G there must exist some set $S\subseteq V$ corresponding to a violated constraint of (IP). In particular the set of all the remaining vertices $S'(=V-F)$ must be always such a set. So, PD increases the dual variable $y_{S'}$ as much as possible until for some vertex u in S' the dual constraint for u becomes tight; i.e.,

$$\sum_{S:u\in S} f^{\mathrm{d}}_S(\delta_S(u))y_S = w_u. \tag{1}$$

Notice that $y_{S'}$ here can indeed be increased because S' is the collection of all those vertices whose corresponding dual constraints were not yet tight. PD adds u into a solution set F and at the same time removes it from S'. Clearly, F eventually becomes an FVS in G (and a feasible solution to (IP)) while y is kept feasible to (D). Lastly, the vertices in F are examined one by one, in the reverse order of their inclusion to F, and whenever any of them is found to be extraneous it is thrown out of F.

3. Performance analysis

The analysis of PD shows that its performance can be estimated by a combinatorial bound of the following form.

Theorem 4. *The algorithm* PD *finds an FVS F for any hypergraph G. The approximation ratio of F is bounded by*

$$\max\left\{\frac{\sum_{u\in X} f_S^{\mathrm{d}}(\delta_S(u))}{f_S^{\mathrm{d}}(E[S])}\right\},$$

where max *is taken over any minimal FVS X in a subgraph $G[S]$ of G induced by any S.*

Proof. From the way PD works it can be seen that PD constructs an FVS F for G and a solution y feasible to (D) simultaneously. The costs of these two solutions are related, by means of (1), in such a way that

$$\sum_{u\in F} w_u = \sum_{u\in F}\sum_{S:u\in S} f_S^{\mathrm{d}}(\delta_S(u))y_S = \sum_{S\subseteq V}\left(\sum_{u\in S\cap F} f_S^{\mathrm{d}}(\delta_S(u))\right) y_S.$$

Compare this with the dual objective function of (D) term by term. Then it follows from the weak duality theorem of LP that the approximation ratio is bounded by the maximum ratio between $\sum_{u\in S\cap F} f_S^{\mathrm{d}}(\delta_S(u))$ and $f_S^{\mathrm{d}}(E[S])$ for any S with non-zero y_S. Thus it remains to show that $S\cap F$ is always a minimal FVS in $G[S]$ whenever $y_S>0$. Let S_l be S' chosen by PD at the lth iteration (of **While**-loop). Notice that $S_l = V - \{u_1,\dots,u_{l-1}\}$ and they are the only S's with $y_S>0$. Recall the last clean-up step (**For**-loop) of the algorithm which examines if u_l's are needed in F in the decreasing order of l. Suppose that for some $j\geqslant 1$, $F\cap S_l$ is a minimal solution in $G[S_l]$ for all $l>j$. If u_j is discarded from F, clearly $F\cap S_l$ is a minimal solution in $G[S_l]$ for all $l\geqslant j$. Assume not and u_j is found non-redundant in F. At this point of time $F\supseteq\{u_1,\dots,u_{j-1}\}$, and hence, this means that $(F-\{u_j\})\cap S_j$ is not an FVS in $G[S_j]$. Therefore, $F\cap S_j$ must be a minimal FVS in $G[S_j]$. Besides, since $u_j\notin S_l$ for all $l>j$, $F\cap S_l$ remains the same as before, a minimal FVS in $G[S_l]$, for all $l>j$. □

In what follows when it is clear from the context on which graph G, $f_{V(G)}^{\mathrm{d}}$ is defined, it will be denoted simply as f^{d}. Observe that in general

$$r(\overline{E}) - r(\overline{E}-\overline{F}) = \#\text{ of edges in } \overline{F} \text{ that must belong to every base of}$$

$$M=(\overline{E},r)$$

for any $\overline{F} \subseteq \overline{E}$, and hence, we may write

$$\begin{aligned} f^{\mathrm{d}}(F) &= \sigma(F) - (f(E) - f(E-F)) = \sigma(F) - (r(\overline{E}) - r(\overline{E-F})) \\ &= \sigma(F) - (r(\overline{E}) - r(\overline{E} - \overline{F})) \\ &= \sigma(F) - (\#\text{ of edges in } \overline{F} \text{ that must belong to every base of} \\ &\quad M = (\overline{E}, r)). \end{aligned} \tag{2}$$

Lemma 5. *Let $X \subseteq V$ be any minimal FVS in a k-hypergraph $G = (V, E)$. Then,*

$$\sum_{u \in X} f^{\mathrm{d}}(\delta(u)) \leqslant k f^{\mathrm{d}}(E).$$

Proof. Suppose G is not connected. Let C_1 be a component of G and C_2 be the rest of G. Then, since both f and σ are additive over components, for any $F \subseteq E$,

$$\begin{aligned} f_V^{\mathrm{d}}(F) &= \sigma(F) + f(E) - f(E-F) \\ &= (\sigma(F \cap E(C_1)) + \sigma(F \cap E(C_2))) + (f(E(C_1)) + f(E(C_2))) \\ &\quad - (f(E(C_1) - (F \cap E(C_1))) + f(E(C_2) - (F \cap E(C_2)))) \\ &= f_{E(C_1)}^{\mathrm{d}}(F \cap E(C_1)) + f_{E(C_2)}^{\mathrm{d}}(F \cap E(C_2)). \end{aligned}$$

Besides, if X is a minimal FVS for G, $X \cap V(C_1)$ must be a minimal FVS for C_1. Thus, it suffices to prove the inequality per component of G. Also notice that it becomes trivial if X consists of a single vertex v for then, $\sum_{u \in X} f^{\mathrm{d}}(\delta(u)) = f^{\mathrm{d}}(\delta(v)) \leqslant f^{\mathrm{d}}(E)$.

So assume henceforth that G is connected and $|X| \geqslant 2$. Let c denote the number of components in $G[V-X]$ and $x = |X|$. Since $E[V-X]$ is acyclic, $r(\overline{E[V-X]}) = \sigma(E[V-X])$. Also, a spanning tree of $\overline{G}$ can be formed by first forming a spanning forest of $\overline{G[V-X]}$ and then connecting it together with the vertices of X. Thus, we may write $f(E) = r(\overline{E}) = r(\overline{E[V-X]}) + (x + c - 1) = \sigma(E[V-X]) + (x + c - 1)$, and hence,

$$\begin{aligned} f^{\mathrm{d}}(E) &= \sigma(E) - f(E) = \sigma(E) - (\sigma(E[V-X]) + x + c - 1) \\ &= \sigma(E[X] \cup E[X, V-X]) - (x + c - 1). \end{aligned}$$

Let $\rho(u)$ denote the number of edges in $\overline{\delta(u)}$ which must belong to every base of M. Then using (2), $f^{\mathrm{d}}(\delta(u)) = \sigma(\delta(u)) - \rho(u)$, and hence,

$$\begin{aligned} k f^{\mathrm{d}}(E) - \sum_{u \in X} f^{\mathrm{d}}(\delta(u)) &= k\sigma(E[X] \cup E[X, V-X]) - \sum_{u \in X} \sigma(\delta(u)) \\ &\quad + \sum_{u \in X} \rho(u) - k(x + c - 1). \end{aligned}$$

Let us now classify the edges of $E[X] \cup E[X, V-X]$ according to their sizes and counts of incidences with X. We say that an edge e is of type $\langle i, j \rangle$ iff $|e| = i$ and it is incident with j vertices of X (i.e. $|e \cap X| = j$), and denote the number of $\langle i,j \rangle$-edges (in $E[X] \cup E[X, V - X]$) by e_{ij}. Since the contribution of any $\langle i,j \rangle$-edge in $k\sigma(E[X] \cup E[X, V-X]) - \sum_{u \in X} \sigma(\delta(u))$ is $k(i-1) - j(i-1) = (k-j)(i-1)$, we may write

$$k\sigma(E[X] \cup E[X, V - X]) - \sum_{u \in X} \sigma(\delta(u)) = \sum_{1 \leqslant j \leqslant i \leqslant k} (k - j)(i - 1)e_{ij}$$

and hence, it suffices to show

$$kf^{\mathrm{d}}(E) - \sum_{u \in X} f^{\mathrm{d}}(\delta(u)) = \sum_{1 \leqslant j \leqslant i \leqslant k} (k - j)(i - 1)e_{ij} + \sum_{u \in X} \rho(u) - k(x + c - 1) \geqslant 0.$$

We shall prove a slightly stronger inequality restricting ourselves to the edges of $E[X, V - X]$ only (i.e. $j < i$). Observe also that

$$\begin{aligned}(k - j)(i - 1) &= ki + j - k - ij = (ki + j - kj - i) + (kj + i - k - ij)\\ &= (k - 1)(i - j) + (k - i)(j - 1) \geqslant (k - 1)(i - j)\end{aligned}$$

for all i and j. So it is clearly sufficient for our purpose to prove

$$\sum_{1 \leqslant j < i \leqslant k} (k - 1)(i - j)e_{ij} + \sum_{u \in X} (\rho(u) - k) - kc \geqslant 0. \tag{3}$$

Assign $-k$ units of potential to each component of $G[V - X]$, $\rho(u) - k$ units to each vertex of X, and $(k - 1)(i - j)$ units to each $\langle i,j \rangle$-edge of $E[X, V - X]$. To prove the validity of (3) we shall appropriately redistribute those potential assigned on edges, vertices, and components, and argue that there remains no one in deficit at the end.

First, take any $\langle i,j \rangle$-edge $e \in E[X, V - X]$ and we distribute its potential evenly to those vertices of $V - X$ incident with e. Since e has $(k - 1)(i - j)$ units of potential and there are exactly $i - j$ vertices in $e \cap (V - X)$, each one receives $k - 1$ units.

Next, pass all the potential assigned this way to the vertices of $V - X$ to the components of $G[V - X]$ they belong to. Observe that at this point any component is no longer in deficit if it contains at least two such vertices since $2(k - 1) - k = k - 2 \geqslant 0$. Even if it is deficient, it is so only by one unit since any component must be incident with at least one edge of $E[X, V - X]$. Let $\mathscr{D}_u$ denote the set of those deficient components adjacent to a vertex u of X.

We now consider how to estimate the value of $\rho(u)$ for any $u \in X$. Let us first define the set $\mathscr{B}_u$ of components of $G[V - X]$ s.t. a component C is in $\mathscr{B}_u$ iff every edge (of $E[X, V - X]$) incident with C contains u (i.e., in $\delta(u)$). Observe now that, to connect those components of $\mathscr{B}_u$ together with the rest of $\overline{G}$, any spanning tree T of $\overline{G}$ can use edges from $\overline{\delta(u)}$ only. This only requires $|\mathscr{B}_u|$ edges of $\overline{\delta(u)}$, and one more edge is needed in T from $\overline{\delta(u)}$ for, otherwise, T cannot be connected (recall $|X| \geqslant 2$). Hence, we can estimate that $\rho(u) \geqslant |\mathscr{B}_u| + 1$. Notice furthermore that the definitions for $\mathscr{B}_u$ and $\mathscr{D}_u$ imply the relationship $\mathscr{D}_u \subseteq \mathscr{B}_u$ for any u. Pass one unit of $\rho(u)$ to each component of $\mathscr{D}_u$ for every $u \in X$, and then, there no longer remains a deficient component. So, still in deficit in G are the vertices of X only, and it is so by $k-\rho(u)+|\mathscr{D}_u|\leqslant k-1-|\mathscr{B}_u-\mathscr{D}_u|$ units.

Recall now that X is a minimal FVS, which implies that u together with $V-X$ induces a cycle for each u of X. A moment of reflection tells us that then such a cycle must be formed by u, a single component C of $G[V-X]$, and either a single $\langle i,1\rangle$-edge $e\in E[X,V-X]$ s.t. $e\cap X=\{u\}$ and $|e\cap V(C)|\geqslant 2$, or two $\langle i,1\rangle$-edges $e_1,e_2\in E[X,V-X]$ s.t. $e_1\cap X=e_2\cap X=\{u\}$ and $|e_1\cap V(C)|=|e_2\cap V(C)|=1$ for some i. Let C_u denote the component of $G[V-X]$ involved in such a construct for $u\in X$. Notice here that while C_u could be chosen by some other vertices of X as their counterparts (i.e., $C_v=C_u$ for $v\neq u$), C_u is never in $\mathscr{D}_u$. In fact C_u is given $2(k-1)$ units exclusively for each $v\in X$ with $C_v=C_u$; thus, if C_u is shared by l vertices of X, it has a surplus of at least $2(k-1)l-k\geqslant k-2$ units. Consider the case when $l\geqslant 2$ and observe that

$$\frac{2(k-1)l-k}{l}=k-2+\left(1-\frac{1}{l}\right)k\geqslant k-2+\frac{1}{2}k\geqslant k-1$$

for $k\geqslant 2$. Thus, C_u has an enough amount of surplus to cover all the deficits at v for v with $C_v=C_u$. So the remaining case is when $l=1$ (i.e., there is no $v\neq u$ s.t. $C_v=C_u$).

Case $C_u\in\mathscr{B}_u$: Then, $|\mathscr{B}_u|\geqslant|\mathscr{D}_u|+1$ and now, u is deficient by at most $k-1-|\mathscr{B}_u-\mathscr{D}_u|\leqslant k-2$ units, which can be collected from the surplus of C_u.

Case $C_u\notin\mathscr{B}_u$: Then, at least $(k-1)$ more units are added to the surplus of C_u by an edge not in $\delta(u)$, sufficient to cover the deficit of u. □

Since a subgraph of a k-hypergraph is always a k-hypergraph, we may apply Lemma 5 to the bound given in Theorem 4, to conclude with the following.

Corollary 6. *The FVS problem for k-hypergraphs can be approximated by* PD *within a factor of k.*

4. Final remarks

It was shown that the FVS problem for k-hypergraphs can be approximated with a factor of k. To illuminate this result with additional insights into the problem structure it is worth pointing out the following facts concerning the approximability of node-deletion problems. The *node-deletion* is a generic problem of finding a minimum weight vertex set whose deletion from a given graph results in a graph satisfying some fixed property π. The VC (FVS, resp.) problem corresponds to the one for the property $\pi=$ "graph has no edges" ("graph is acyclic", resp.). All the node-deletion problems for such hereditary properties are known to be NP-complete [14], and so far, excluding rather trivial cases as given in [15], the constant factor approximable cases are known only for properties of very special structure [7]; that is, such a property π that in any graph G the edge sets of subgraphs of G satisfying π form a matroid. Indeed, the property $\pi=$ "acyclic" for FVS, for instance, "induces" a matroid on any graph

G (namely, the cycle matroid of G). On the other hand, the system induced by the same π on a hypergraph is no longer a matroid, and in fact the problem of finding a maximum acyclic subgraph in a k-hypergraph is much harder when $k \geqslant 3$; it becomes quite involved already with $k = 3$ (called the *matroid matching* problem [13]), generalizing both matroid intersection and general graph matching, and it becomes NP-hard for $k \geqslant 4$. The result of this paper thus shows that the constant factor approximation for a node-deletion problem can coexist with the hardness in the corresponding *maximum subgraph* (or *edge-deletion*) problem.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, Proof verification and hardness of approximation problems, Proceedings of the 33rd FOCS, IEEE, Los Alamitos, 1992, pp. 14–23.
[2] V. Bafna, P. Berman, T. Fujito, Constant ratio approximations of the weighted feedback vertex set problem for undirected graphs, Proceedings of the sixth ISAAC, Lecture Notes in Computer Science, vol. 1004, Springer, Berlin, 1995, pp. 142–151.
[3] R. Bar-Yehuda, S. Even, A local-ratio theorem for approximating the weighted vertex cover problem, Annals of Discrete Mathematics, vol. 25, North-Holland, Amsterdam, 1985, pp. 27–46.
[4] R. Bar-Yehuda, D. Geiger, J. Naor, R.M. Roth, Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and Bayesian inference, Proceedings of the fifth SODA, ACM, New York, 1994, pp. 344–354.
[5] A. Becker, D. Geiger, Approximation algorithms for the loop cutset problem, Proceedings of the 10th Conference on Uncertainty in AI, Morgan Kaufman, 1994, pp. 60–68.
[6] P. Erdős, L. Pósa, On the maximal number of disjoint circuits of a graph, Publ. Math. Debrechen 9 (1962) 3–12.
[7] T. Fujito, A primal-dual approach to approximation of node-deletion problems for matroidal properties, Proceedings of 24th ICALP, Lecture Notes in Computer Science, vol. 1256, Springer, Berlin, 1997, pp. 749–759.
[8] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.
[9] J. Håstad, Some optimal in-approximability results, Proceedings of the 29th STOC, ACM, New York, 1997, pp. 1–10.
[10] D.S. Hochbaum, Approximation algorithms for the set covering and vertex cover problems, SIAM J. Comput. 11(3) (1982) 555–556.
[11] D.H. Hochbaum (Ed.), Approximation Algorithm for NP-Hard Problems, PWS, Boston, 1996.
[12] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.
[13] L. Lovász, The matroid matching problem, Algebraic Methods in Graph Theory, vol. 2, North-Holland, Amsterdam, 1981, pp. 495–518.
[14] J.M. Lewis, M. Yannakakis, The node-deletion problem for hereditary properties is NP-complete, J. Comput. System Sci. 20 (1980) 219–230.
[15] C. Lund, M. Yannakakis, The approximation of maximum subgraph problems, Proceedings of the 20th ICALP, Lecture Notes in Computer Science, vol. 700, Springer, Berlin, 1993, pp. 40–51.
[16] B. Monien, R. Schulz, Four approximation algorithms for the feedback vertex set problem, Proceedings of the seventh Conference on Graph Theoretic Concepts of Computer Science, 1981, pp. 315–326.
[17] B. Monien, E. Speckenmeyer, Ramsey numbers and an approximation algorithm for the vertex cover problem, Acta Inform. 22 (1985) 115–123.
[18] C. Papadimitriou, M. Yannakakis, Optimization, approximation and complexity classes, J. Comput. System Sci. 43 (1991) 425–440.